EXHIBIT 16
UNITED STATES PATENT NO. 7,532,808
CLAIM CHART FOR INFRINGEMENT OF CLAIM 7 BY ASUS ACCUSED PRODUCTS

As demonstrated in the chart below, ASUS directly and indirectly infringes at least claim 7 of US 7,532,808 (the "'808 Patent"). ASUS directly infringes, contributes to the infringement of, and/or induces infringement of the '808 Patent by making, using, selling, offering for sale, and/or importing into the United States the Accused Products that are covered by at least claim 7 of the '808 Patent. The Accused Products are devices that decode H.264-compliant video. For example, ASUS Q543MV Notebook ("ASUS Q543MV") is a representative product for other ASUS devices that decode H.264-compliant video.

The ASUS Q543MV contains at least one video decoder that helps decode H.264-compliant video.[1] While evidence from the ASUS Q543MV is specifically charted herein, the evidence and contentions charted herein apply equally to the other ASUS Accused Products that decode H.264-compliant video.

No part of this exemplary chart construes, or is intended to construe, the specification, file history, or claims of the '808 Patent. Moreover, this exemplary chart does not limit, and is not intended to limit, Nokia's infringement positions or contentions.

The following infringement chart includes exemplary citations to ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audiovisual services (available at https://www.itu.int/rec/T-REC-H.264-200503-S/en) (the "H.264 Standard"). The cited functionality has been included in editions of the H.264 Standard since at least May 2003 and remains in current editions of the H.264 Standard. Any ASUS device that includes a decoder that practices the functionality in any of these editions of the H.264 Standard ("H.264 Decoder") practices at least claim 7 of the '808 Patent.

Nokia contends each of the following limitations is met literally, and, to the extent a limitation is not met literally, it is met under the doctrine of equivalents. [2]

[1] See, e.g., https://www.asus.com/us/laptops/for-home/everyday-use/asus-vivobook-pro-15-oled-q543/techspec/; https://www.intel.com/content/www/us/en/products/sku/236849/intel-core-ultra-9-processor-185h-24m-cache-up-to-5-10-ghz/specifications.html; https://developer.nvidia.com/video-encode-and-decode-gpu-support-matrix-new.

[2] This claim chart is based on the information currently available to Nokia and is intended to be exemplary in nature. Nokia reserves all rights to update and elaborate its infringement positions, including as Nokia obtains additional information during the course of discovery.

EXHIBIT 16
UNITED STATES PATENT NO. 7,532,808
CLAIM CHART FOR INFRINGEMENT OF CLAIM 7 BY ASUS ACCUSED PRODUCTS

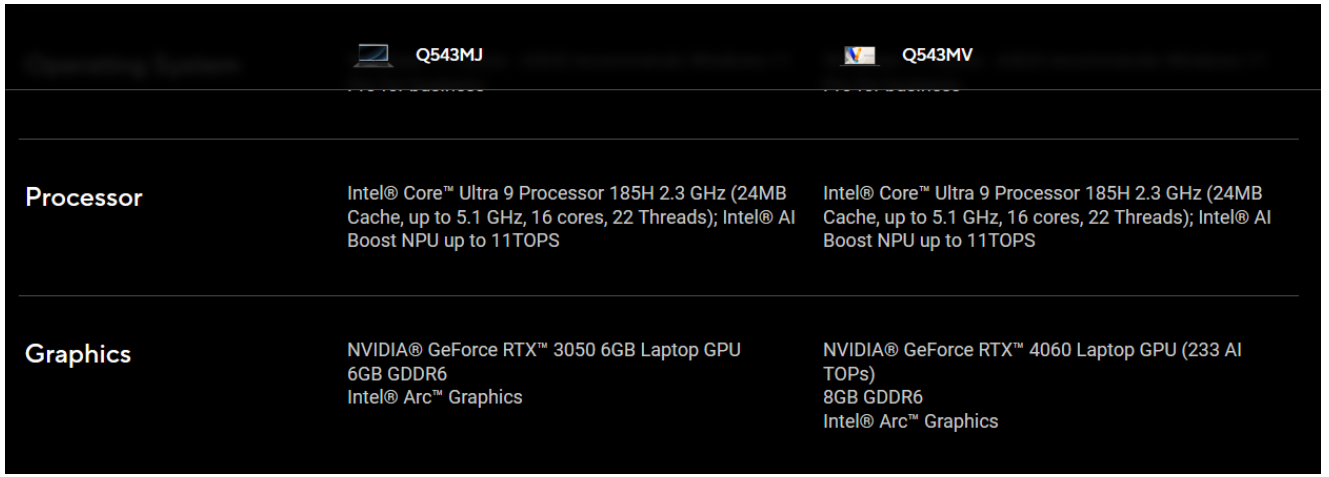
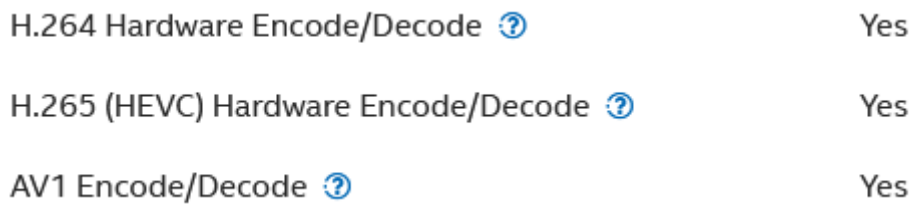
U.S. PATENT NO. 7,532,808	ASUS ACCUSED PRODUCTS
7. [A] A method of decoding an encoded video sequence, the method comprising:	Each of the Accused Products, such as the ASUS Q543MV, performs a method of decoding an encoded video sequence. For example, and without limitation, the Asus Q543MV uses hardware-accelerated decoding and includes an NVIDIA GeForce RTX 4060 Laptop graphics processing unit ("GPU") and an Intel Core Ultra 9 Processor 185H.  Source: https://www.asus.com/us/laptops/for-home/everyday-use/asus-vivobook-pro-15-oled-q543/techspec/ (last accessed March 6, 2025). 

EXHIBIT 16
UNITED STATES PATENT NO. 7,532,808
CLAIM CHART FOR INFRINGEMENT OF CLAIM 7 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 7,532,808	ASUS ACCUSED PRODUCTS
	Source: https://www.intel.com/content/www/us/en/products/sku/236849/intel-core-ultra-9-processor-185h-24m-cache-up-to-5-10-ghz/specifications.html (last accessed March 6, 2025)(specifications for Intel Core Ultra 9 185H).

↕ BOARD	↕ FAMILY	↕ NVENC Generation	Desktop/ Mobile	# OF CHIPS	Total # of NVENC	Max # of concurrent sessions	H.264 (AVCHD) YUV 4:2:0	H.264 (AVCHD) YUV 4:2:2	H.264 (AVCHD) YUV 4:4:4	H.264 (AVCHD) Lossless	H.265 (HEVC) 4K YUV 4:2:0	H.265 (HEVC) YUV 4:2:2	H.265 (HEVC) 4K YUV
GeForce RTX 4060 Laptop	Ada Lovelace	8th Gen	M	1	1	8	YES	NO	YES	YES	YES	NO	YES
GeForce RTX 4060	Ada Lovelace	8th Gen	D	1	1	8	YES	NO	YES	YES	YES	NO	YES

↕ BOARD	↕ FAMILY	↕ NVDEC Generation	Desktop/ Mobile	# OF CHIPS	Total # of NVDEC	MPEG-1	MPEG-2	VC-1	VP8	VP9 4:2:0			H.264 (AVCHD) 4:2:0	
										8 Bit	10 Bit	12 Bit	8 Bit	10 Bit
GeForce RTX 4060 Laptop	Ada Lovelace	5th Gen	M	1	1	YES	YES	YES	YES	YES	YES	YES	YES	NO

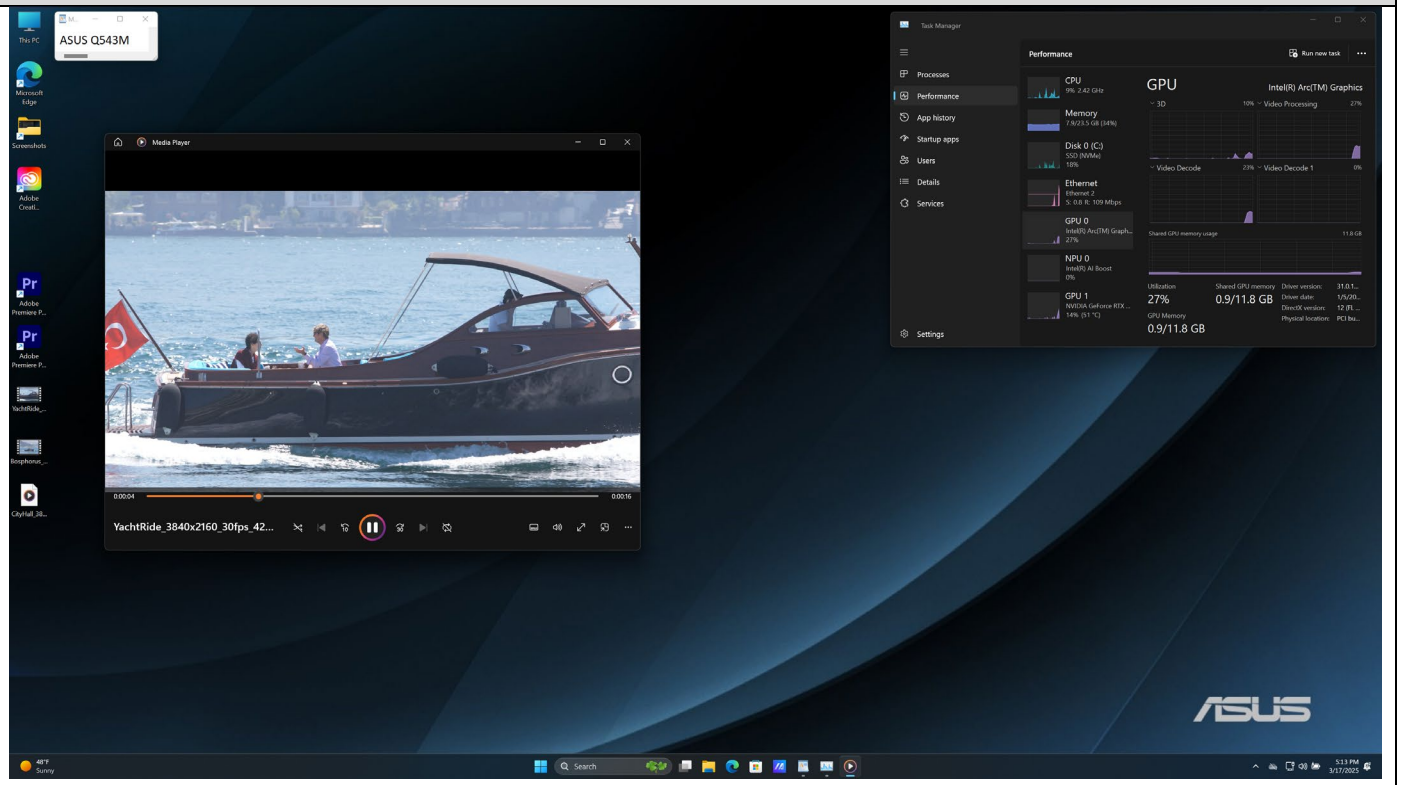
H.265 (HEVC) 4:2:0			H.265 (HEVC) 4:2:2		H.265 (HEVC) 4:4:4			AV1	
8 Bit	10 Bit	12 Bit	8 Bit	10 Bit	8 Bit	10 Bit	12 Bit	8 Bit	10 Bit
YES	YES	YES	NO	NO	YES	YES	YES	YES	YES

Source: https://developer.nvidia.com/video-encode-and-decode-gpu-support-matrix-new (last accessed March 6, 2025) (row for 4060 Laptup GPU).

For example, an ASUS Q543MV was used to playback an H.264-compliant video.

EXHIBIT 16
UNITED STATES PATENT NO. 7,532,808
CLAIM CHART FOR INFRINGEMENT OF CLAIM 7 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 7,532,808	ASUS ACCUSED PRODUCTS



Source: Screenshot of H.264-compliant video playback on ASUS Q543MV.

The following specifications provide further evidence of how the Accused Products operate:

> **0.2 Purpose**
>
> . . .
>
> This Recommendation | International Standard was developed in response to the growing need for higher compression of moving pictures for various applications such as videoconferencing, digital storage media, television broadcasting, internet streaming, and communication. It is also designed

EXHIBIT 16
UNITED STATES PATENT NO. 7,532,808
CLAIM CHART FOR INFRINGEMENT OF CLAIM 7 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 7,532,808	ASUS ACCUSED PRODUCTS
	to enable the use of the coded video representation in a flexible manner for a wide variety of network environments. The use of this Recommendation | International Standard allows motion video to be manipulated as a form of computer data and to be stored on various storage media, transmitted and received over existing and future networks and distributed on existing and future broadcasting channels. (ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audio visual services at p. 1). **3 Definitions** For the purposes of this Recommendation | International Standard, the following definitions apply. . . . **3.1 access unit**: A set of *NAL units* always containing exactly one *primary coded picture*. In addition to the *primary coded picture*, an access unit may also contain one or more *redundant coded pictures* or other *NAL units* not containing *slices* or *slice data partitions* of a *coded picture*. The decoding of an access unit always results in a *decoded picture*. . . . **3.14 bitstream**: A sequence of bits that forms the representation of *coded pictures* and associated data forming one or more *coded video sequences*. Bitstream is a collective term used to refer either to a *NAL unit stream* or a *byte stream*. . . . **3.27 coded picture**: A *coded representation* of a *picture* . . . . . . **3.30 coded video sequence**: A sequence of *access units* that consists, in decoding order, of an *IDR access unit* followed by zero or more non-IDR *access units* including all subsequent *access units* up to but not including any subsequent *IDR access unit*. . . . **3.37 decoded picture:** A *decoded picture* is derived by decoding a *coded picture*. A *decoded picture* is either a decoded *frame*, or a decoded *field*. A decoded *field* is either a decoded *top field* or a decoded *bottom field*. . . .

EXHIBIT 16
UNITED STATES PATENT NO. 7,532,808
CLAIM CHART FOR INFRINGEMENT OF CLAIM 7 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 7,532,808	ASUS ACCUSED PRODUCTS
	3.41 decoding process: The process specified in this Recommendation | International Standard that reads a *bitstream* and derives *decoded pictures* from it. . . . **3.109 primary coded picture:** The coded representation of a *picture* to be used by the *decoding process* for a bitstream conforming to this Recommendation | International Standard. The primary coded picture contains all *macroblocks* of the *picture.* The only *pictures* that have a normative effect on the *decoding process* are primary coded pictures. See also *redundant coded picture.* . . . (ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audio visual services at pp. 4–6).
[B] receiving an indication of a skip coding mode for a first segment;	Each of the Accused Products, such as the ASUS Q543MV, performs a method of decoding an encoded video sequence, the method comprising receiving an indication of a skip coding mode for a first segment. For example, and without limitation, the H.264 Standard specifies the following regarding the decoding process. For example, as defined in Subclause 7.3.4, the Slice Layer Syntax includes a syntax element mb_skip_run (when CAVLC entropy coding is used) or mb_skip_flag (in the case of CABAC entropy coding). For P and SP slices, for example, a value of mb_skip_flag equal to 1 indicates that a macroblock has been coded as a skipped macroblock or, correspondingly, a value of mb_skip_run greater than zero indicates that a number of consecutive macroblocks has been coded as skipped macroblocks (see definition in Subclause 3.135). Upon receiving such an indication, an H.264-compliant decoder infers that mb_type for the current macroblock is P_Skip (Table 7-13). The following specifications provide further evidence of how the Accused Products operate: **3 Definitions** For the purposes of this Recommendation | International Standard, the following definitions apply. . . . **3.15 block**: An MxN (M-column by N-row) array of samples, or an MxN array of *transform coefficients*.

EXHIBIT 16
UNITED STATES PATENT NO. 7,532,808
CLAIM CHART FOR INFRINGEMENT OF CLAIM 7 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 7,532,808	ASUS ACCUSED PRODUCTS
	. . . **3.24 chroma**: An adjective specifying that a sample array or single sample is representing one of the two colour difference signals related to the primary colours. The symbols used for a chroma array or sample are Cb and Cr. . . . **3.74 luma**: An adjective specifying that a sample array or single sample is representing the monochrome signal related to the primary colours. The symbol or subscript used for luma is Y or L. . . . **3.75 macroblock**: A 16x16 *block* of *luma* samples and two corresponding *blocks* of *chroma* samples . . . **3.135 skipped macroblock**: A *macroblock* for which no data is coded other than an indication that the *macroblock* is to be decoded as "skipped". This indication may be common to several *macroblocks*. . . . (ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audio visual services at pp. 8–11).

EXHIBIT 16
UNITED STATES PATENT NO. 7,532,808
CLAIM CHART FOR INFRINGEMENT OF CLAIM 7 BY ASUS ACCUSED PRODUCTS

7.3.4 Slice data syntax

slice_data( ) {	C	Descriptor
if(entropy_coding_mode_flag)		
while(!byte_aligned( ))		
cabac_alignment_one_bit	2	f(1)
CurrMbAddr = first_mb_in_slice * (1 + MbaffFrameFlag)		
moreDataFlag = 1		
prevMbSkipped = 0		
do {		
if(slice_type != I && slice_type != SI)		
if(!entropy_coding_mode_flag) {		
mb_skip_run	2	ue(v)
prevMbSkipped = (mb_skip_run > 0)		
for(i=0; i<mb_skip_run; i++)		
CurrMbAddr = NextMbAddress(CurrMbAddr)		
moreDataFlag = more_rbsp_data( )		
} else {		
mb_skip_flag	2	ae(v)
moreDataFlag = !mb_skip_flag		
}		

(ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audio visual services at p. 49).

7.4.4 Slice data semantics

. . .

mb_skip_run specifies the number of consecutive skipped macroblocks for which, when decoding a P or SP slice, mb_type shall be inferred to be P_Skip and the macroblock type is collectively referred to as a P macroblock type . . .

. . .

EXHIBIT 16
UNITED STATES PATENT NO. 7,532,808
CLAIM CHART FOR INFRINGEMENT OF CLAIM 7 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 7,532,808	ASUS ACCUSED PRODUCTS
	mb_skip_flag equal to 1 specifies that for the current macroblock, when decoding a P or SP slice, mb_type shall be inferred to be P_Skip and the macroblock type is collectively referred to as P macroblock type . . . . . . (ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audio visual services at p. 85). **7.4.5 Macroblock layer semantics** **mb_type** specifies the macroblock type. The semantics of mb_type depend on the slice type. . . . Macroblock types that may be collectively referred to as P macroblock types are specified in Table 7-13. . . . **Table 7-13 – Macroblock type values 0 to 4 for P and SP slices**

EXHIBIT 16
UNITED STATES PATENT NO. 7,532,808
CLAIM CHART FOR INFRINGEMENT OF CLAIM 7 BY ASUS ACCUSED PRODUCTS

<table>
<tr><th>U.S. PATENT NO. 7,532,808</th><th>ASUS ACCUSED PRODUCTS</th></tr>
<tr><td></td><td>
<table>
<tr><th>mb_type</th><th>Name of mb_type</th><th>NumMbPart (mb_type)</th><th>MbPartPredMode (mb_type, 0)</th><th>MbPartPredMode (mb_type, 1)</th><th>MbPartWidth (mb_type)</th><th>MbPartHeight (mb_type)</th></tr>
<tr><td>0</td><td>P_L0_16x16</td><td>1</td><td>Pred_L0</td><td>na</td><td>16</td><td>16</td></tr>
<tr><td>1</td><td>P_L0_L0_16x8</td><td>2</td><td>Pred_L0</td><td>Pred_L0</td><td>16</td><td>8</td></tr>
<tr><td>2</td><td>P_L0_L0_8x16</td><td>2</td><td>Pred_L0</td><td>Pred_L0</td><td>8</td><td>16</td></tr>
<tr><td>3</td><td>P_8x8</td><td>4</td><td>na</td><td>na</td><td>8</td><td>8</td></tr>
<tr><td>4</td><td>P_8x8ref0</td><td>4</td><td>na</td><td>na</td><td>8</td><td>8</td></tr>
<tr><td>inferred</td><td>P_Skip</td><td>1</td><td>Pred_L0</td><td>na</td><td>16</td><td>16</td></tr>
</table>

The following semantics are assigned to the macroblock types in Table 7-13.
. . .
– P_Skip: no further data is present for the macroblock in the bitstream.
. . .
(ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audio visual services at pp. 85, 88–89).
</td></tr>
<tr><td>**[C]** assigning either a zero motion vector or a predicted non-zero motion vector for the skip coding mode for the first segment based at least in part on the motion</td><td>Each of the Accused Products, such as the ASUS Q543MV, performs a method of decoding an encoded video sequence, the method comprising assigning either a zero motion vector or a predicted non-zero motion vector for the skip coding mode for the first segment based at least in part on the motion information of a second segment neighboring the first segment.</td></tr>
</table>

EXHIBIT 16
UNITED STATES PATENT NO. 7,532,808
CLAIM CHART FOR INFRINGEMENT OF CLAIM 7 BY ASUS ACCUSED PRODUCTS

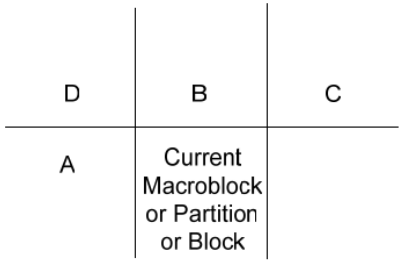
U.S. PATENT NO. 7,532,808	ASUS ACCUSED PRODUCTS
information of a second segment neighboring the first segment; and	For example, and without limitation, the H.264 Standard specifies the following regarding the decoding process and H.264-compliant bitstream. For example, the H.264/AVC Standard describes the derivation process for luma motion vectors for skipped macroblocks in P and SP slices in Subclause 8.4.1.1. Subclause 8.4.1.1 specifies that depending on the values of motion vectors in the neighbouring blocks to the left of the current macroblock and above the current macroblock, the current macroblock will be assigned either a zero-value or a non-zero value motion vector. By way of example, for a non-zero luma motion vector, the decoder predicts its value based on motion data of neighbouring blocks. The following specifications provide further evidence of how the Accused Products operate: **6.4.8 Derivation processes for neighbouring macroblocks, blocks, and partitions** . . . Figure 6-14 illustrates the relative location of the neighbouring macroblocks, blocks, or partitions A, B, C, and D to the current macroblock, partition, or block, when the current macroblock, partition, or block is in frame coding mode.  **Figure 6-14 – Determination of the neighbouring macroblock, blocks, and partitions (informative)** (ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audio visual services at pp. 28–29) **8.4.1 Derivation process for motion vector components and reference indices** Inputs to this process are

EXHIBIT 16
UNITED STATES PATENT NO. 7,532,808
CLAIM CHART FOR INFRINGEMENT OF CLAIM 7 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 7,532,808	ASUS ACCUSED PRODUCTS
	– a macroblock partition mbPartIdx, – a sub-macroblock partition subMbPartIdx. Outputs of this process are – luma motion vectors mvL0 and mvL1 as well as the chroma motion vectors mvCL0 and mvCL1 – reference indices refIdxL0 and refIdxL1 – prediction list utilization flags predFlagL0 and predFlagL1 – a sub-partition macroblock motion vector count variable subMvCnt For the derivation of the variables mvL0 and mvL1 as well as refIdxL0 and refIdxL1, the following applies. – If mb_type is equal to P_Skip, the derivation process for luma motion vectors for skipped macroblocks in P and SP slices in subclause 8.4.1.1 is invoked with the output being the luma motion vectors mvL0 and reference indices refIdxL0, and predFlagL0 is set equal to 1. mvL1 and refIdxL1 are marked as not available and predFlagL1 is set equal to 0. The sub-partition motion vector count variable subMvCnt is set equal to 1. – Otherwise, if mb_type is equal to B_Skip or B_Direct_16x16 or sub_mb_type[mbPartIdx] is equal to B_Direct_8x8, the derivation process for luma motion vectors for B_Skip, B_Direct_16x16, and B_Direct_8x8 in B slices in subclause 8.4.1.2 is invoked with mbPartIdx and subMbPartIdx as the input and the output being the luma motion vectors mvL0, mvL1, the reference indices refIdxL0, refIdxL1, the sub-partition motion vector count subMvCnt, and the prediction utilization flags predFlagL0 and predFlagL1. – Otherwise, for X being replaced by either 0 or 1 in the variables predFlagLX, mvLX, refIdxLX, and in Pred_LX and in the syntax elements ref_idx_lX and mvd_lX, the following applies. . . . **8.4.1.1 Derivation process for luma motion vectors for skipped macroblocks in P and SP slices** This process is invoked when mb_type is equal to P_Skip.

EXHIBIT 16
UNITED STATES PATENT NO. 7,532,808
CLAIM CHART FOR INFRINGEMENT OF CLAIM 7 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 7,532,808	ASUS ACCUSED PRODUCTS
	Outputs of this process are the motion vector mvL0 and the reference index refIdxL0. The reference index refIdxL0 for a skipped macroblock is derived as follows. refIdxL0 = 0. For the derivation of the motion vector mvL0 of a P_Skip macroblock type, the following applies. – The process specified in subclause 8.4.1.3.2 is invoked with mbPartIdx set equal to 0, subMbPartIdx set equal to 0, currSubMbType set equal to "na", and listSuffixFlag set equal to 0 as input and the output is assigned to mbAddrA, mbAddrB, mvL0A, mvL0B, refIdxL0A, and refIdxL0B. – The variable mvL0 is specified as follows. – If any of the following conditions are true, both components of the motion vector mvL0 are set equal to 0. – mbAddrA is not available – mbAddrB is not available – refIdxL0A is equal to 0 and both components of mvL0A are equal to 0 – refIdxL0B is equal to 0 and both components of mvL0B are equal to 0 – Otherwise, the derivation process for luma motion vector prediction as specified in subclause 8.4.1.3 is invoked with mbPartIdx = 0, subMbPartIdx = 0, refIdxL0, and currSubMbType = "na" as inputs and the output is assigned to mvL0. NOTE – The output is directly assigned to mvL0, since the predictor is equal to the actual motion vector. (ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audio visual services at pp. 137–39). **8.4.1.3 Derivation process for luma motion vector prediction** Inputs to this process are – the macroblock partition index mbPartIdx, – the sub-macroblock partition index subMbPartIdx,

EXHIBIT 16
UNITED STATES PATENT NO. 7,532,808
CLAIM CHART FOR INFRINGEMENT OF CLAIM 7 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 7,532,808	ASUS ACCUSED PRODUCTS
	– the reference index of the current partition refIdxLX (with X being 0 or 1), – the variable currSubMbType. Output of this process is the prediction mvpLX of the motion vector mvLX (with X being 0 or 1). The derivation process for the neighbouring blocks for motion data in subclause 8.4.1.3.2 is invoked with mbPartIdx, subMbPartIdx, currSubMbType, and listSuffixFlag = X (with X being 0 or 1 for refIdxLX being refIdxL0 or refIdxL1, respectively) as the input and with mbAddrN\mbPartIdxN\subMbPartIdxN, reference indices refIdxLXN and the motion vectors mvLXN with N being replaced by A, B, or C as the output. The derivation process for median luma motion vector prediction in subclause 8.4.1.3.1 is invoked with mbAddrN\mbPartIdxN\subMbPartIdxN, mvLXN, refIdxLXN with N being replaced by A, B, or C and refIdxLX as the input and mvpLX as the output, unless one of the following is true. – MbPartWidth(mb_type) is equal to 16, MbPartHeight(mb_type) is equal to 8, mbPartIdx is equal to 0, and refIdxLXB is equal to refIdxLX, mvpLX = mvLXB – MbPartWidth(mb_type) is equal to 16, MbPartHeight(mb_type) is equal to 8, mbPartIdx is equal to 1, and refIdxLXA is equal to refIdxLX, mvpLX = mvLXA – MbPartWidth(mb_type) is equal to 8, MbPartHeight(mb_type) is equal to 16, mbPartIdx is equal to 0, and refIdxLXA is equal to refIdxLX, mvpLX = mvLXA – MbPartWidth(mb_type) is equal to 8, MbPartHeight(mb_type) is equal to 16, mbPartIdx is equal to 1, and refIdxLXC is equal to refIdxLX, mvpLX = mvLXC . . .

EXHIBIT 16
UNITED STATES PATENT NO. 7,532,808
CLAIM CHART FOR INFRINGEMENT OF CLAIM 7 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 7,532,808	ASUS ACCUSED PRODUCTS
	8.4.1.3.1 Derivation process for median luma motion vector prediction Inputs to this process are – the neighbouring partitions mbAddrN\mbPartIdxN\subMbPartIdxN (with N being replaced by A, B, or C), – the motion vectors mvLXN (with N being replaced by A, B, or C) of the neighbouring partitions, – the reference indices refIdxLXN (with N being replaced by A, B, or C) of the neighbouring partitions, and – the reference index refIdxLX of the current partition. Output of this process is the motion vector prediction mvpLX. The variable mvpLX is derived as follows: – When both partitions mbAddrB\mbPartIdxB\subMbPartIdxB and mbAddrC\mbPartIdxC\subMbPartIdxC are not available and mbAddrA\mbPartIdxA\subMbPartIdxA is available, mvLXB = mvLXA mvLXC = mvLXA refIdxLXB = refIdxLXA refIdxLXC = refIdxLXA – Depending on reference indices refIdxLXA, refIdxLXB, or refIdxLXC, the following applies. – If one and only one of the reference indices refIdxLXA, refIdxLXB, or refIdxLXC is equal to the reference index refIdxLX of the current partition, the following applies. Let refIdxLXN be the reference index that is equal to refIdxLX, the motion vector mvLXN is assigned to the motion vector prediction mvpLX: mvpLX = mvLXN

EXHIBIT 16
UNITED STATES PATENT NO. 7,532,808
CLAIM CHART FOR INFRINGEMENT OF CLAIM 7 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 7,532,808	ASUS ACCUSED PRODUCTS
	– Otherwise, each component of the motion vector prediction mvpLX is given by the median of the corresponding vector components of the motion vector mvLXA, mvLXB, and mvLXC: mvpLX[0] = Median(mvLXA[0], mvLXB[0], mvLXC[0]) mvpLX[1] = Median(mvLXA[1], mvLXB[1], mvLXC[1]) **8.4.1.3.2 Derivation process for motion data of neighbouring partitions** Inputs to this process are – the macroblock partition index mbPartIdx, – the sub-macroblock partition index subMbPartIdx, – the current sub-macroblock type currSubMbType, – the list suffix flag listSuffixFlag Outputs of this process are (with N being replaced by A, B, or C) – mbAddrN\mbPartIdxN\subMbPartIdxN specifying neighbouring partitions, – the motion vectors mvLXN of the neighbouring partitions, and – the reference indices refIdxLXN of the neighbouring partitions. Variable names that include the string "LX" are interpreted with the X being equal to listSuffixFlag. The partitions mbAddrN\mbPartIdxN\subMbPartIdxN with N being either A, B, or C are derived in the following ordered steps. 1. Let mbAddrD\mbPartIdxD\subMbPartIdxD be variables specifying an additional neighbouring partition. 2. The process in subclause 6.4.8.5 is invoked with mbPartIdx, currSubMbType, and subMbPartIdx as input and the output is assigned to mbAddrN\mbPartIdxN\subMbPartIdxN with N being replaced by A, B, C, or D. 3. When the partition mbAddrC\mbPartIdxC\subMbPartIdxC is not available, the following applies

EXHIBIT 16
UNITED STATES PATENT NO. 7,532,808
CLAIM CHART FOR INFRINGEMENT OF CLAIM 7 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 7,532,808	ASUS ACCUSED PRODUCTS
	mbAddrC = mbAddrD mbPartIdxC = mbPartIdxD subMbPartIdxC = subMbPartIdxD The motion vectors mvLXN and reference indices refIdxLXN (with N being A, B, or C) are derived as follows. – If the macroblock partition or sub-macroblock partition mbAddrN\mbPartIdxN\subMbPartIdxN is not available or mbAddrN is coded in Intra prediction mode or predFlagLX of mbAddrN\mbPartIdxN\subMbPartIdxN is equal to 0, both components of mvLXN are set equal to 0 and refIdxLXN is set equal to –1. – Otherwise, the following applies. – The motion vector mvLXN and reference index refIdxLXN are set equal to MvLX[mbPartIdxN][subMbPartIdxN] and RefIdxLX[mbPartIdxN], respectively, which are the motion vector mvLX and reference index refIdxLX that have been assigned to the (sub-)macroblock partition mbAddrN\mbPartIdxN\subMbPartIdxN. – The variables mvLXN[1] and refIdxLXN are further processed as follows. – If the current macroblock is a field macroblock and the macroblock mbAddrN is a frame macroblock mvLXN[1] = mvLXN[1] / 2 refIdxLXN = refIdxLXN * 2 – Otherwise, if the current macroblock is a frame macroblock and the macroblock mbAddrN is a field macroblock mvLXN[1] = mvLXN[1] * 2 refIdxLXN = refIdxLXN / 2 – Otherwise, the vertical motion vector component mvLXN[1] and the reference index refIdxLXN remain unchanged. (ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audio visual services at pp. 148–49).

EXHIBIT 16
UNITED STATES PATENT NO. 7,532,808
CLAIM CHART FOR INFRINGEMENT OF CLAIM 7 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 7,532,808	ASUS ACCUSED PRODUCTS
[D] forming a prediction for the first segment with respect to a reference frame based at least in part on the assigned motion vector for the skip coding mode, wherein the assigned motion vector is one of the zero motion vector and the predicted non-zero motion vector.	Each of the Accused Products, such as the ASUS Q543MV, performs a method of decoding an encoded video sequence, the method comprising forming a prediction for the first segment with respect to a reference frame based at least in part on the assigned motion vector for the skip coding mode, wherein the assigned motion vector is one of the zero motion vector and the predicted non-zero motion vector. For example, and without limitation, the H.264/AVC Standard specifies the decoding process for Inter prediction samples in Subclause 8.4.2. The following specifications provide further evidence of how the Accused Products operate: **8.4 Inter prediction process** This process is invoked when decoding P and B macroblock types. Outputs of this process are Inter prediction samples for the current macroblock that are a 16x16 array predL of luma samples and when chroma_format_idc is not equal to 0 (monochrome) two 8x8 arrays predCr and predCb of chroma samples, one for each of the chroma components Cb and Cr. The partitioning of a macroblock is specified by mb_type. Each macroblock partition is referred to by mbPartIdx. When the macroblock partitioning consists of partitions that are equal to sub-macroblocks, each sub-macroblock can be further partitioned into sub-macroblock partitions as specified by sub_mb_type. Each sub-macroblock partition is referred to by subMbPartIdx. When the macroblock partitioning does not consist of sub-macroblocks, subMbPartIdx is set equal to 0. . . . The Inter prediction process for a macroblock partition mbPartIdx and a sub-macroblock partition subMbPartIdx consists of the following ordered steps . . . 1. Derivation process for motion vector components and reference indices as specified in subclause 8.4.1.

EXHIBIT 16
UNITED STATES PATENT NO. 7,532,808
CLAIM CHART FOR INFRINGEMENT OF CLAIM 7 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 7,532,808	ASUS ACCUSED PRODUCTS
	Inputs to this process are – a macroblock partition mbPartIdx, – a sub-macroblock partition subMbPartIdx. Outputs of this process are – luma motion vectors mvL0 and mvL1 and when chroma_format_idc is not equal to 0 (monochrome) the chroma motion vectors mvCL0 and mvCL1 – reference indices refIdxL0 and refIdxL1 – prediction list utilization flags predFlagL0 and predFlagL1 – the sub-macroblock partition motion vector count subMvCnt. 2. The variable MvCnt is incremented by subMvCnt. 3. Decoding process for Inter prediction samples as specified in subclause 8.4.2. Inputs to this process are – a macroblock partition mbPartIdx, – a sub-macroblock partition subMbPartIdx. – variables specifying partition width and height for luma and chroma (if available), partWidth, partHeight, partWidthC (if available), and partHeightC (if available) – luma motion vectors mvL0 and mvL1 and when chroma_format_idc is not equal to 0 (monochrome) the chroma motion vectors mvCL0 and mvCL1 – reference indices refIdxL0 and refIdxL1 – prediction list utilization flags predFlagL0 and predFlagL1 Outputs of this process are – inter prediction samples (pred); which are a (partWidth)x(partHeight) array predPartL of prediction luma samples and when chroma_format_idc is not equal to 0 (monochrome) two (partWidthC)x(partHeightC) arrays predPartCr, and predPartCb of prediction chroma samples, one for each of the chroma components Cb and Cr. . . . (ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audio visual services at pp. 135–36).

EXHIBIT 16
UNITED STATES PATENT NO. 7,532,808
CLAIM CHART FOR INFRINGEMENT OF CLAIM 7 BY ASUS ACCUSED PRODUCTS

U.S. PATENT NO. 7,532,808	ASUS ACCUSED PRODUCTS
	8.4.2 Decoding process for Inter prediction samples Inputs to this process are – a macroblock partition mbPartIdx, – a sub-macroblock partition subMbPartIdx. – variables specifying partition width and height for luma and chroma (if available), partWidth, partHeight, partWidthC (if available) and partHeightC (if available) – luma motion vectors mvL0 and mvL1 and when chroma_format_idc is not equal to 0 (monochrome) chroma motion vectors mvCL0 and mvCL1 – reference indices refIdxL0 and refIdxL1 Outputs of this process are – the Inter prediction samples predPart, which are a (partWidth)x(partHeight) array predPartL of prediction luma samples, and when chroma_format_idc is not equal to 0 (monochrome) two (partWidthC)x(partHeightC) arrays $predPart_{Cb}$, $predPart_{Cr}$ of prediction chroma samples, one for each of the chroma components Cb and Cr. . . . (ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audio visual services at pp. 149–50).